

Integrity

From the ITGS Guide:

Integrity refers to the correspondence of data with itself, at its creation. Data lacks integrity when it has been changed accidentally or tampered with. For example, a hacker might change driver license data resulting in arrests of innocent people.

Data integrity is concerned with the 'correctness' of the data. Data cannot be regarded as having integrity if it has been accidentally changed or tampered with. Data that lacks integrity cannot be trusted but is hard to spot. Errors may be introduced into data in a variety of ways. They can be introduced when the person typing in the data misreads it off a source document or if a program or machine errors corrupt data. Some types of corruption can be caused by simple typing errors.

Validation and verification checks are performed on data to ensure its integrity.

Verification

When keying-in very large quantities of data it is normal for some errors to occur in copying from the hand-written source documents. Mistakes made when copying are known as **transcription errors**. It is also quite common for people to **transpose** characters, so that 69 would become 96, for example. To eliminate copying errors the data is re-typed by a second key-to-disk operator and any differences in the two sets of data are notified by the computer. Checking and correcting errors made when keying-in data from source documents onto disk or tape is known as **verification**.

Validation

Validation checks are intended to ensure that the data is suitable for the purpose for which it is being used, in particular that the data is:

- Of the correct type (alphabetic, numeric etc.)
- Within an acceptable numerical range

- Complete
- In the right format

Validation checks must be combined with verification checks to reduce the number of errors in data input and ensure the **integrity** (accuracy and completeness) of the data.

Validation *type* checks examples:

- 6-digit STD code: acceptable 028373
- Rejected 0392Z5 (alphabetic character included)
- car registration: acceptable ACE 3641
- rejected 8CE 3641 (number instead of letter at the start)

Validation *range* checks examples:

- school students/pupils ages: $5 \leq \text{age} \leq 19$
- examination percentages: $0 \leq \text{marks} \leq 100$

****Cross site Scripting***

Introduction

Websites today are more complex than ever, containing a lot of dynamic content making the experience for the user more enjoyable. Dynamic content is achieved through the use of web

applications which can deliver different output to a user depending on their settings and needs. Dynamic websites suffer from a threat that static websites don't, called "Cross Site Scripting" (or XSS dubbed by other security professionals). Currently small informational tidbits about Cross Site Scripting holes exist but none really explain them to an average person or administrator. This FAQ was written to provide a better understanding of this emerging threat, and to give guidance on detection and prevention.

"What is Cross Site Scripting?"

Cross site scripting (also known as XSS) occurs when a web application gathers malicious data from a user. The data is usually gathered in the form of a hyperlink which contains malicious content within it. The user will most likely click on this link from another website, instant message, or simply just reading a web board or email message. Usually the attacker will encode the malicious portion of the link to the site in HEX (or other encoding methods) so the request is less suspicious looking to the user when clicked on. After the data is collected by the web application, it creates an output page for the user containing the malicious data that was originally sent to it, but in a manner to make it appear as valid content from the website. Many popular guestbook and forum programs allow users to submit posts with html and javascript embedded in them. If for example I was logged in as "john" and read a message by "joe" that contained malicious javascript in it, then it may be possible for "joe" to hijack my session just by reading his bulletin board post. Further details on how attacks like this are accomplished via "cookie theft" are explained in detail below.

"What does XSS and CSS mean?"

Often people refer to Cross Site Scripting as CSS. There has been a lot of confusion with Cascading Style Sheets (CSS) and cross site scripting. Some security people refer to Cross Site Scripting as XSS. If you hear someone say "I found a XSS hole", they are talking about Cross Site Scripting for certain.

"What are the threats of Cross Site Scripting?"

Often attackers will inject JavaScript, VBScript, ActiveX, HTML, or Flash into a vulnerable application to fool a user (Read below for further details) in order to gather data from them. Everything from account hijacking, changing of user settings, cookie theft/poisoning, or false advertising is possible. New malicious uses are being found every day for XSS attacks. The post below by Brett Moore brings up a good point with regard to "Denial Of Service", and potential "auto-attacking" of hosts if a user simply reads a post on a message board.

****Acunetix**

Audit your website security with Acunetix Web Vulnerability Scanner

Website security is possibly today's most overlooked aspect of securing the enterprise and should be a priority in any organization. Hackers are concentrating their efforts on web-based applications - shopping carts, forms, login pages, dynamic content, etc. Web applications are accessible 24 hours a day, 7 days a week and control valuable data since they often have direct access to backend data such as customer databases.

Firewalls, SSL and locked-down servers are futile against web application hacking

Any defense at network security level will provide no protection against web application attacks since they are launched on port 80 - which has to remain open. In addition, web applications are often tailor-made therefore tested less than off-the-shelf software and are more likely to have undiscovered vulnerabilities. **Acunetix WVS automatically checks your web applications for SQL Injection, XSS & other web vulnerabilities.**